

#7

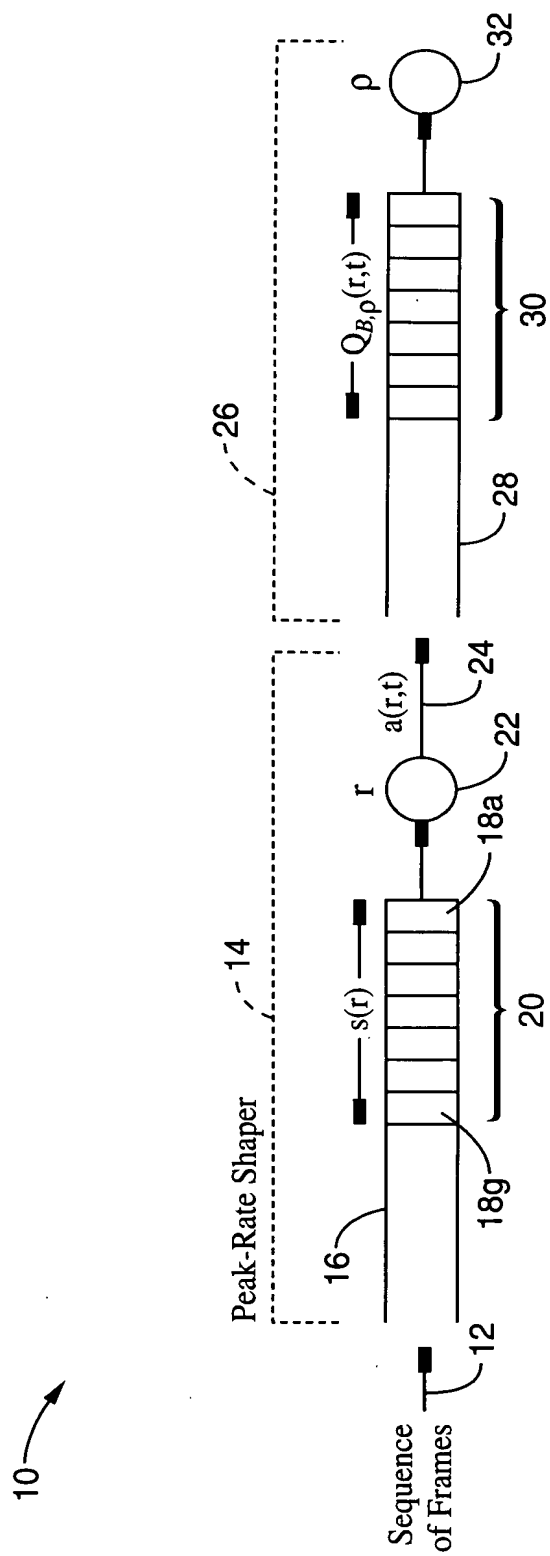


FIG. 1

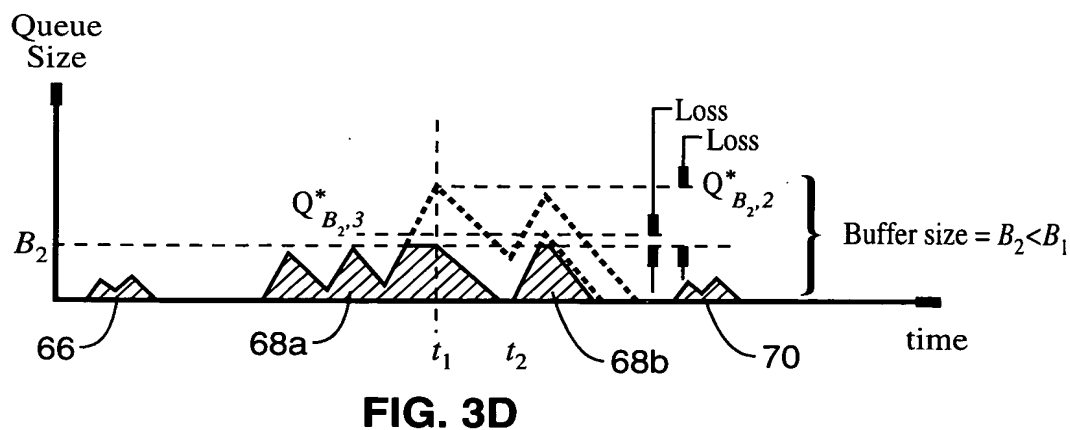
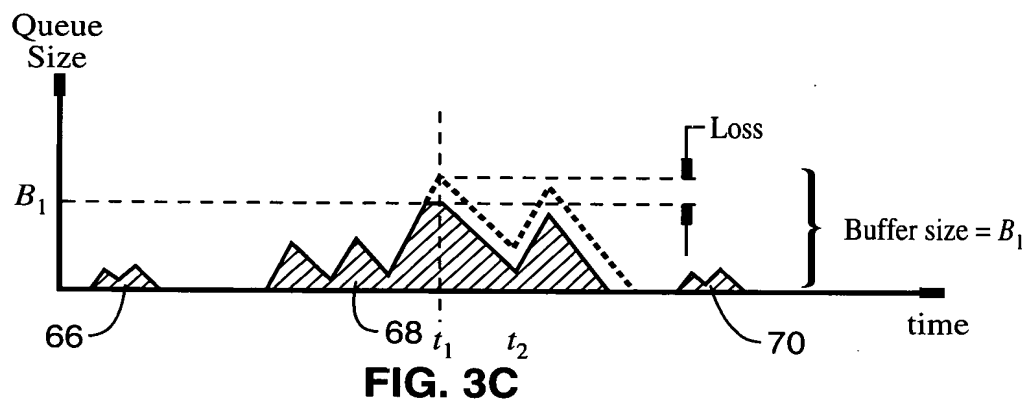
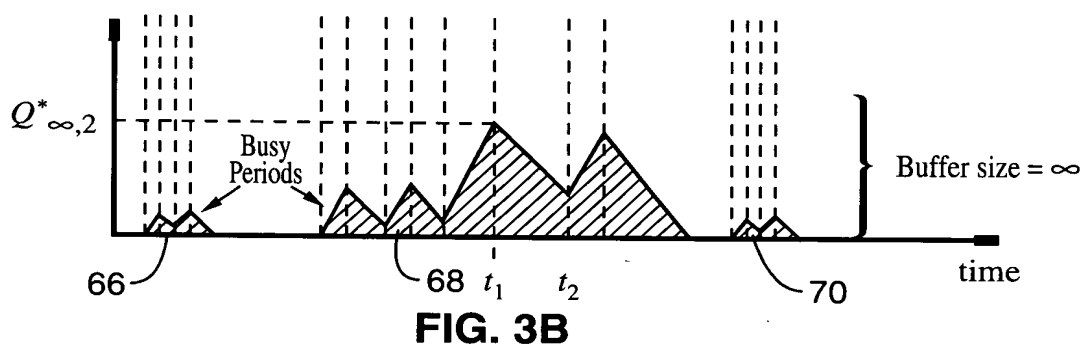
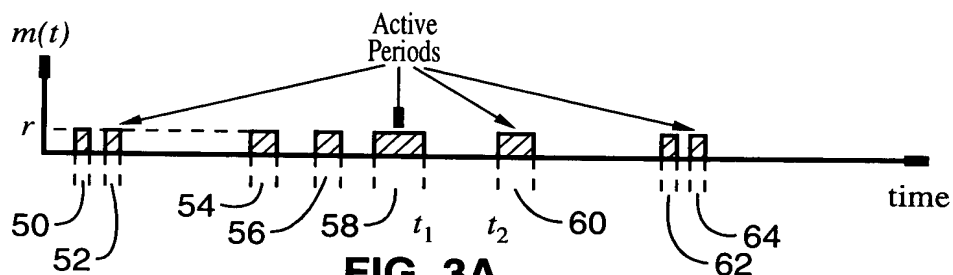
```

/* ===== */
/* ===== Compute Active Periods ===== */
/* ===== for an Elementary Video Stream ===== */
/* ===== */

/* Index  $i$  denotes the  $i$ -th frame,  $q_i$  denotes the queue size at the instant
just after the  $i$ -th frame arrival and index  $j$  denotes the  $j$ -th active period. */
1 /* Perform initialization */
1.1  $s_1^r \leftarrow 1/f$ ;  $q_1 \leftarrow d_1$ ;  $i \leftarrow j \leftarrow 1$ ;
2
2.1 If  $([q_i]/(r)) < \epsilon^1/f$  /* no backlog present at the start of next frame */
2.1.1  $q_{i+1} \leftarrow d_{i+1}$ ;
2.1.2  $t_j^r \leftarrow i/f + [q_i]/(r)$ ; /* compute end time of current active period */
2.1.3  $j \leftarrow j+1$ ;
2.1.4  $s_j^r \leftarrow [(i+1)/(f)]$ ; /* start time of new active period */
2.2 else
2.2.1  $q_{i+1} \leftarrow q_i + d_{i+1} - i/f$ ; /* there is backlog carried to current frame */
2.3 endif
3
3.1  $i \leftarrow i+1$ ;
3.2 If  $(i < N)$ 
3.2.1 goto Step 2;
3.3 endif
4
4.1  $s(r) \leftarrow \max_{1 \leq k \leq N} q_k$ ; /* compute maximum queue length observed */
4.2  $t_j^r \leftarrow N/f + [q_N]/(r)$ ; /* compute end time of last active period */
4.3  $n_a(r) \leftarrow j$ ; /* store the number of active periods */
4.4 Define  $s_{na(r)+1} = \infty$ ;

```

FIG. 2



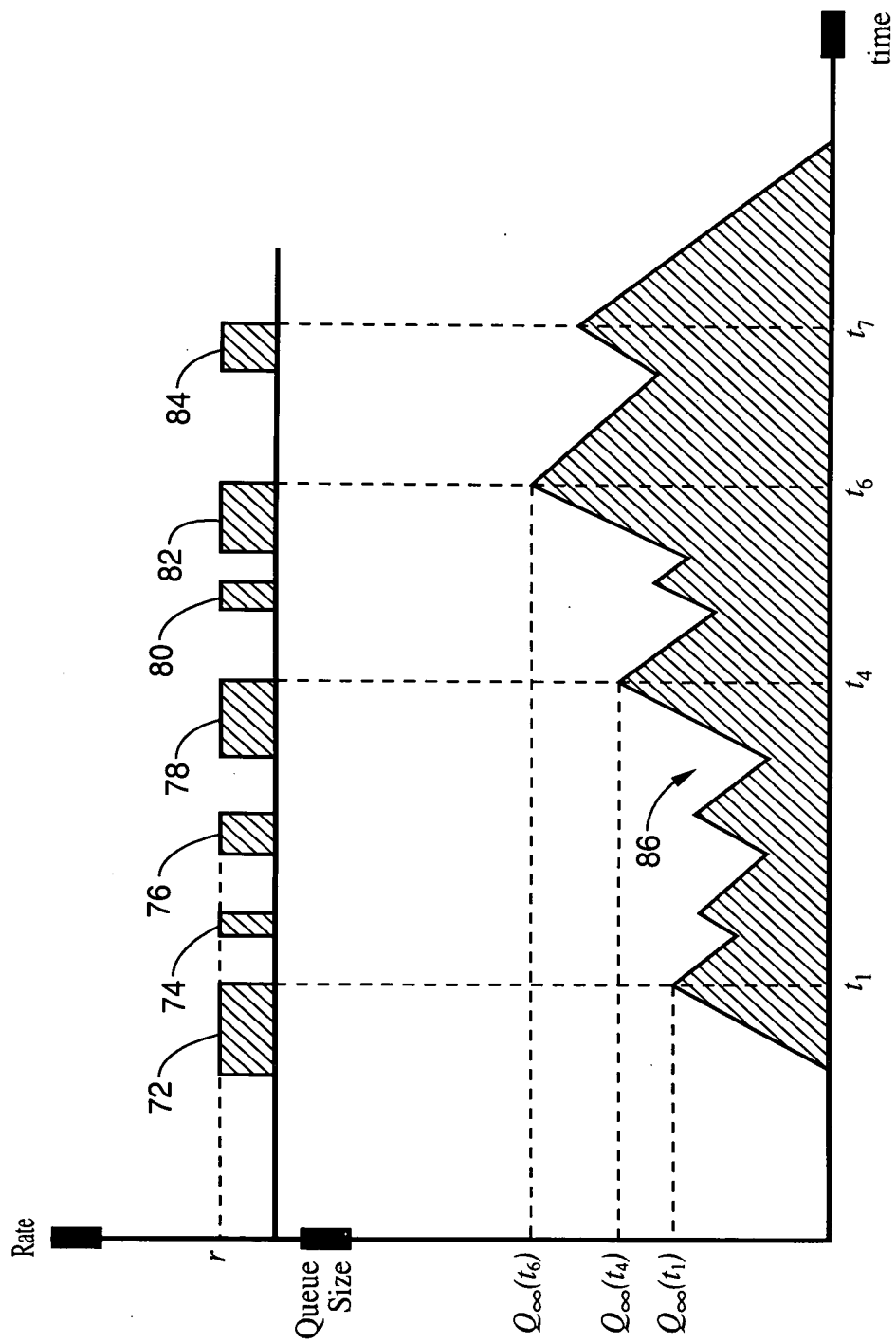


FIG. 4

```

/* ===== */
/* ===== Compute Loss Curve ===== */
/* ===== for an Elementary Video Stream ===== */
/* ===== */

/* INIT.  $B$  holds the available buffer,  $L_B$  holds the sum of the unrestricted queue sizes of
the busy
periods with loss at the time instants at which the maximum queue occurs,  $n_l$  holds the number
of busy periods with loss,  $n_b$  the number of busy periods, and  $n_a$  the number of active periods.*/
1.1  $B \leftarrow \infty$  ;  $L_B \leftarrow 0$ ;  $n_l \leftarrow 0$ ;  $M \leftarrow \text{stream\_size}$ ;
/* Compute busy periods for rate  $\rho$ , and buffer size  $B$ . */
1.2 process_active_periods(1,  $n_a$ ,  $B$ );
/* Insert into the heap the buffer point at which a loss occurs
for each busy period, compute the buffer size for the first break for
each busy period and insert it into the heap as well. */
1.3 For  $b = 1$  to  $n_b$  /* for each busy period  $b$  */
1.3.1 heap_insert( $Q_b^*$ , LOSS,  $b$ );
1.3.2  $B' \leftarrow \text{compute\_next\_break}(b)$ ;
1.3.3 heap_insert( $B'$ , BREAK,  $b$ );
1.4 endfor
2 /* Extract the maximum buffer from the heap, and process
the corresponding busy period until the heap becomes empty. */
2.1 ( $b$ ,  $B$ , cause)  $\leftarrow \text{heap\_extract\_max}()$ ;
/* Let  $(s_p, t_p)$ ,  $(s_{p+1}, t_{p+1})$ , ...,  $(s_q, t_q)$  be the active periods contained within the busy period  $b$ .
Also, let  $(s_p, t_p)$ , ...,  $(s_j, t_j)$  be the active periods contained within the interval  $(\alpha_b, \tau_b)$ . */
2.2 If (cause = LOSS)
/* update  $n_l$  and  $L_B$  variables */
2.2.1  $L_B \leftarrow L_B + (\sum_{i=p}^j r(t_i - s_i) - \rho(t_j - s_p))$ ;
2.2.2  $n_l \leftarrow n_l + 1$ ;
2.3 else /* cause = BREAK */
/* update  $n_l$  and  $L_B$  variables */
2.3.1  $L_B \leftarrow L_B - (\sum_{i=p}^j r(t_i - s_i) - \rho(t_j - s_p))$ ;
2.3.2  $n_l \leftarrow n_l - 1$ ;
/* Process the break in busy period  $b$ : Compute the new busy periods, and update  $n_l$  and  $L_B$ 
when a busy period already experiences loss. For the remaining new busy periods, insert
into the heap the buffer points at which the first losses occur. Finally, for all the new
busy periods, compute the buffer sizes for the first break and insert them into the heap. */
2.3.3 process_break( $b$ ,  $B$ );
2.4 endif
2.5 output_point( $B$ ,  $[(L_B - n_l B) / (M)]$ );
2.6 If (heap not empty)
2.6.1 goto Step 2;
2.7 endif
/* Output the last point of the loss curve */
2.8 Output (0,  $[(r - \rho) / (r)]$ );
2.9 STOP;

```

FIG. 5

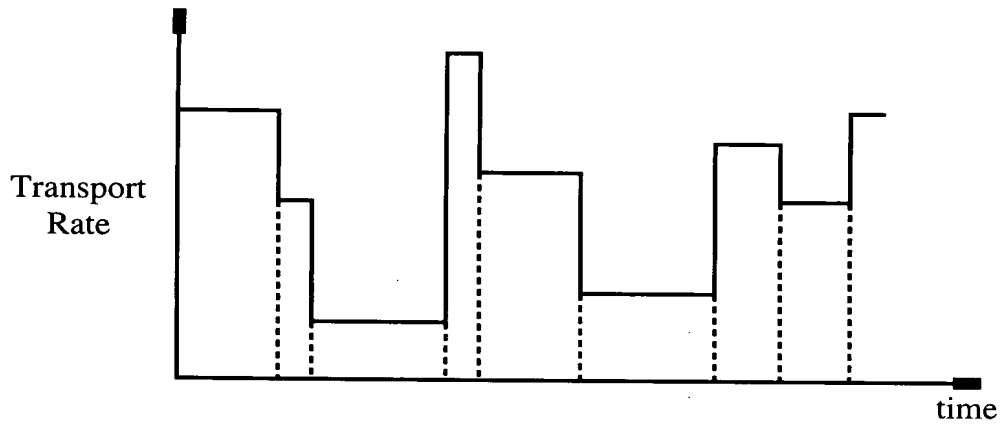


FIG. 6

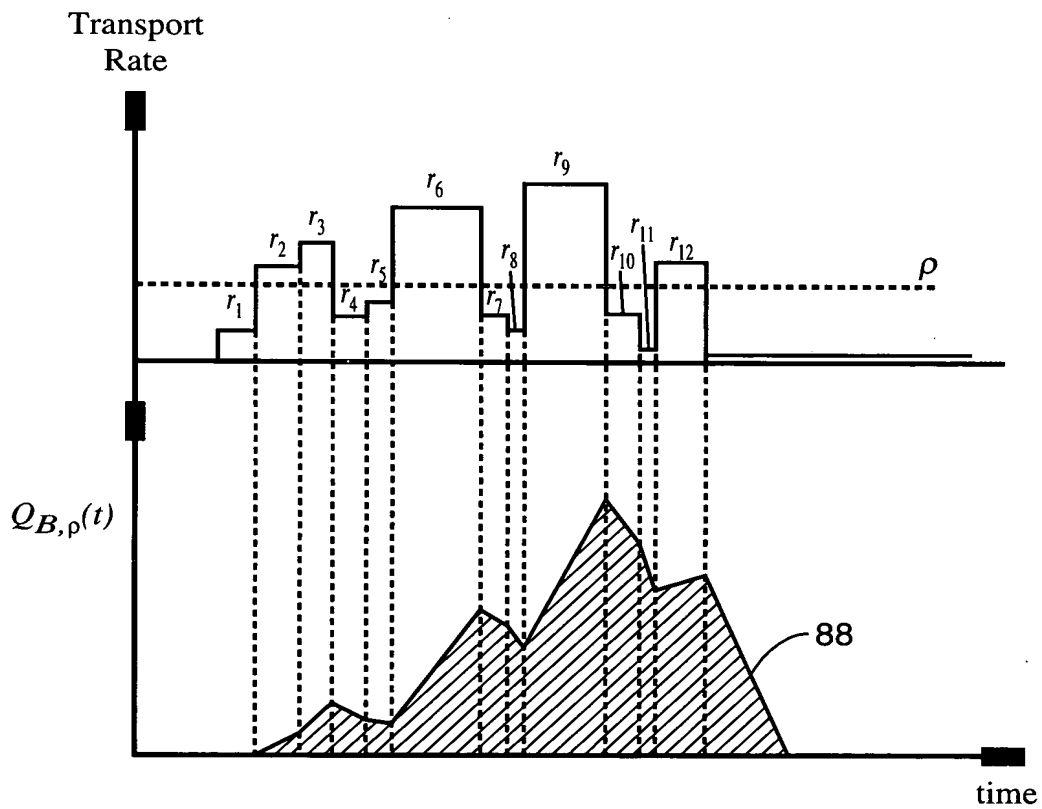


FIG. 7

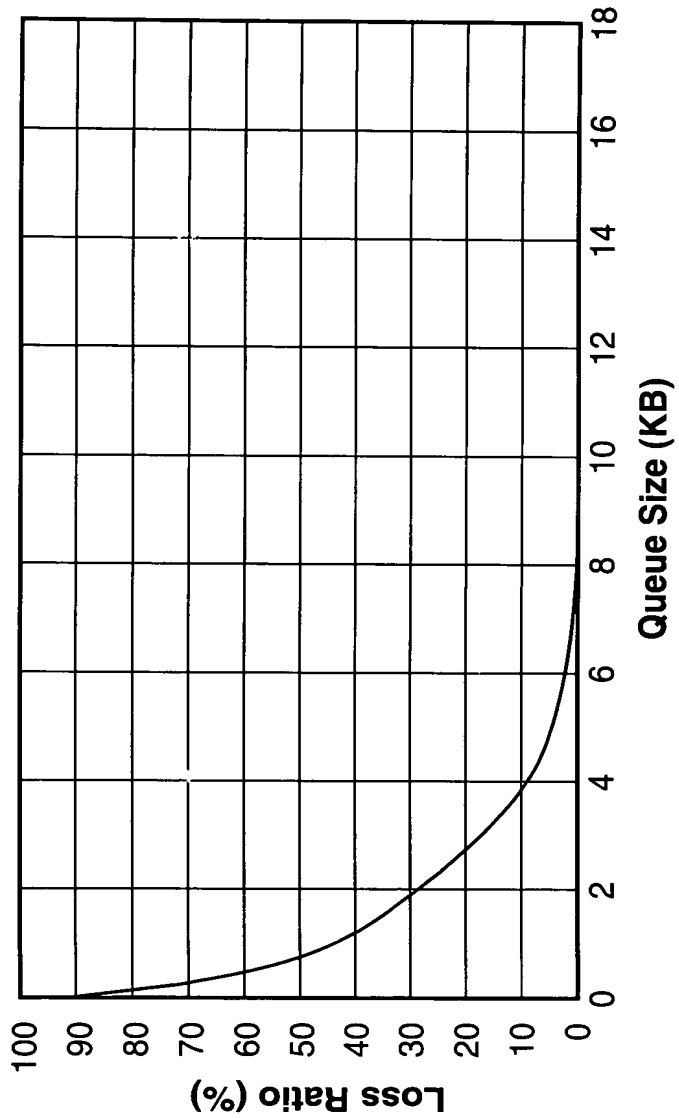


FIG. 8

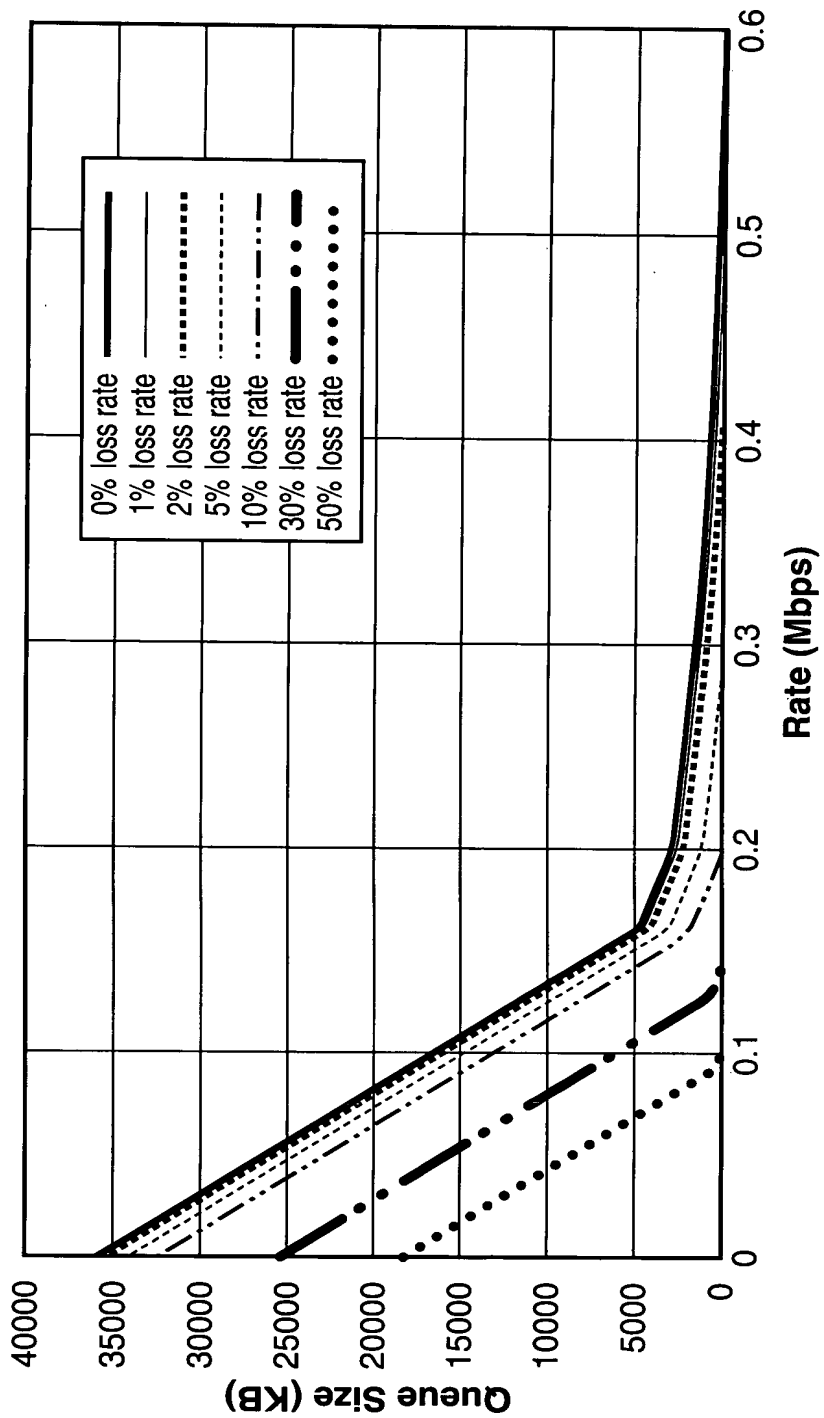


FIG. 9